The Boring Web

Essay / World Wide Web

A fundamental Problem

Yesterday night I worked on the PythonOnliner again in some places and made some notes at the same time. For example, I play with the idea of develope a small web browser. This is anything but difficult in PyQt4 and there are hundreds of tutorials and good documentation on the topic. Now I have asked myself some questions about the design, because there are many points that really bother me with previous browsers.

Most browsers are too overloaded and have super great features that I will never use in my life. Bookmarks, for example, is one of them. I save my links completely oldschool in a text file, because with Waterfox this is simply too fiddly for me and feeds my browser with additional data, which can be sent or read out in the worst case. I don't want to. Although most tools and features can be installed naturally, but they cannot be completely uninstalled. But there are also other functions that bother me, e.g. the history feature. I don't want that, but I can't install it. So my digital footprint is some data bigger again.

I find the function to install addons really practical, you have to trust the developers but with open source software (in the best case) several people already have an eye for possible inconsistencies. Then there is another completely different topic. The presentation of the web has developed in recent years from an information web to an entertainment web. Everything is flashing, has hundreds of different font types, JavaScript (I don't reject it completely, but should be used more sparingly), libraries and all kinds of additional information that are basically not necessary. As you can see on the screenshot, you can also use (html pseudocode)

```
.html
   .pre[content]:pre
:html
```

to display information in a `.html` file. This will certainly not please many technicians and developers who work on web standards and also the browser manufacturers might be a thorn in the side, because then simply less information can be read out, nevertheless the web should work like this.

When we look at a website we primarily want to be able to grasp the content well. Pictures, links and colors are additional information that are still pretty but not really necessary. If you want to read a text, you need a text. This is a fundamental (personal) problem when I surf the web. Everyone wants to convince me that his offer is the best and I always find myself not reading texts because I don't like the design. Maybe the person is very pretty, but I just think his clothes are terrible.

So if I were to develop a small web browser for the PythonOnliner, my focus would be on readability of the content and not on entertainment. Text instead of design. You can certainly solve this with a prefabricated CSS and simply instruct the browser to use only this and no other. There are still many things to consider, but overall I want to document that so that I can sort out the individual ideas and thoughts.

The difference between entertainment and information design

I had to think about it for a few days before I could formulate my thoughts. The web has evolved from a technical patchwork (which I think is very good). Although HTML and CSS have developed good standards from the main languages of web design over the years, designers still have a lot of freedom. Nobody is forbidden to place red font on a green background with blue elements. It becomes quite strenuous for the visitors who want to find information on the website and here already begins the difference. Information and design only fit together when the designer and the creator of the content have come together. This only happens in a few cases and usually we have informative content in an anti-information design.

Below I made a screenshot of fefes Blog. This blog was created by one of the most famous security blogger from Germany. I'm not going to go into the backend technology any further, I just want to discuss the design. There is no CSS installed because the pages are rendered with standard HTML elements (h2, h3, b, ul, li, a, div) and so on. Any browser will be able to read this. There is no design because it is not needed. The aim of this blog is to provide information. The information must be quickly and intuitively recognizable. Everyone who has been on the web more than ten times will perceive the blue underlined font as a link. Because it's always been done this way. It is therefore very likely that we are dealing here with information design.

Now let's look at the exact opposite. The Bandcamp page of Entangled Everthing is a page where you can listen to the current album (or another album from the discography). So you only need one music player with the usual functions. Start, stop, pause and change song. You can put the window in the background and still play the music. The musicians can change a lot of the design of the page, e.g. adjust the font color according to their wishes, insert pictures, further information etc. It should be noted that this information is only *important* for people *if* they need more. I can also just listen to the music without reading anything on the site or buying the album. So it's very probably entertainment design.

The articles in the Paris Review contain information and there is an elaborated web design. Journalistic publications are products that are sold. Felix von Leitner does not have to blog and he earns his money mainly with his work as an IT security expert. The Paris Review, on the other hand, is a magazine with staff and journalists who have to compete with a whole series of online magazines. Online magazines try to enhance their product with design and to sell information and many people like it. If you only want to read a little superficially, that's no problem, it's just superficial entertainment information. With such magazines I am no longer inclined to read the articles if, for example, they have implemented a static header in the design. In some cases I use the Reader View function. So it's very probably entertainment design. Hybrid design is a mixture of information and entertainment design.

K, cool story, bro. What's your problem?

My problem is bad design. I get along very well with information design, also with the exact opposite of entertainment design. Only hybrid design fucks me. The superfluous design disturbs reading. When I format it with browser functions (see screenshot above) or with addons, I have to rest on the tasteless default of my browser. There was a very short period of time when mobile websites were created for so-called future phones. Usually there was only one header with the design of the e.g. New York Times and the articles in a list. You could do that again, you don't have to change your whole offer.

RSS as a microblog system

Most people have already come into contact with RSS, even if they don't know the specifications by heart, they have seen the typical orange logo at least once. RSS has many advantages and disadvantages. RSS is nothing else but a microblog like Mastodon, only the Twitter clone looks prettier. There is a standardization, which makes it easy for developers to work with RSS, so you can for example integrate a feed into your own `.html` project without problems and feed it with information. The target group can then access it with a browser, online services such as feedreader or Inoreader, apps, desktop software and terminal scripts. Up to now, this has been used by many websites as a one-side-channel technique.

An on-side-channel posts a marker or short information about an article that is linked in the entry. Since it is very difficult to monetize a feed or access user data, not much importance has been attached to it in the market economy. This is seen by many *professionals*as a negative point. If we take a closer look at microblogging(a), RSS(b) and blogs(c) we will see some similarities and all three can be defined in the following sentence:

```
(a), (b), and (c) are collections of individual entries of individual
length that are displayed to a reader in chronological order.
```

And overall the syntax is - strenuous - to read. But I have full control over my data, because all information comes into a file and is then uploaded to a server. If I don't like something, I just delete it again. If I want, I can add a comment and perhaps stimulate a discussion with it. at Twitter, Mastodon and other Microblog services I must announce myself. There is always the possibility that an account can be hacked. I can hide a text file somewhere in the depths of my operating system and upload it again if necessary. I can make backups on the internet on secure disposable accounts e.g. at Gitlab. A whole blog, in the worst case scenario with a database, is neither easy to rescue nor can I (in many cases) just pull backup data from it. So far I manage my rss.xml file by editing it with my PythonOnliner. I have written some functions that simplify my work. Nevertheless, it should be more like a software that allows me to manage my RSS channel better. File upload and download. Maybe a more customized syntax that doesn't burn in my eyes after an hour of work (personal opinion) and sends a ping to my readers when I publish a new article. A microblog system based on the KISS (Keep it simple, stupid) principle. The biggest problem so far is that RSS only works in one direction. I post content. Readers read content.

Design. Design never changes

As noted above, the design of rss/xml files in the browser is incredibly ugly (please remember, design is a matter of personal taste). My idea was to manipulate my Waterfox browser to display a new design. It wasn't about creating something release-ready right away, but rather about testing the limits of my browser a little bit. Since I already had a rough design, I didn't have to worry about it. But first the *hacker avatar* image had to disappear from my feed, because it looked childish anyway and I had only used it as a gap filler.

What remained was the feed that most people see in their browsers. You have the typical subscribe

field and the content. Both are separated from each other and can be considered separately. Since I am not a web developer I had to examine my browser a little more closely. After half an hour I started to work with the web developer console and the style editor to do a first test. For an xml file there are three CSS files. First the *global.css*, the subscribe *.css* and the *intl.css*. I simply copied my *hackerbit.css* into the *global.css*. I have not looked at the standards, manuals or the exact guidelines. I just wanted to know if this would work. I had implemented my own branding. In the next step I changed the *subscribe.css* file so that it was no longer displayed at the top of the browser. I pulled the browsers so narrow, because I wanted to know how it looks scaled. Why I did this I will describe in more detail in a later paragraph. I have not changed the *intl.css* file and I saved the changed CSS files to my desktop.

I like the design much better. The ugly scubscribe area is gone (I wonder if someone really uses it, because there is a *subscribe to this page*button in the toolbar). The headings are displayed in the typical yellow `#F18C00` and therefore shine very brightly. Even if one can argue again at this point. Even though the trend in web design is towards brutalist web design and I agree with many points, I find underlined links incredibly tiring. I only work with very few colors and make sure that this is still easy to read after some articles. For me personally, yellow works best.

Overall, my design is structured and follows the function. Everything is clearly arranged and I keep everything as minimalistic as possible. People who visit my website should be able to capture and process the content as quickly as possible. All elements that could interfere with this must be omitted. Less is more. My font is kept in grey #606C76, because the eyes are too strained by black. I follow the minimalistic tradition, which I also practice privately and thus have taken the burden of *wanting to own everything*from me. You have to be able to concentrate on one clearly defined point and if that reaches a website (or the RSS feed) you will achieve more than fancy design and superfluous elements. To test whether the design can also be applied to other RSS feeds, I took a closer look at certain elements.

For example, I don't use a date in my entries because I live in a different time on the web. (I like to compare the digitaltime with the mythological dreamtime of the Indigenous Australians).

```
The dreamtime legends deal with the universal, space- and
timeless world, from which the real present emerges in an
incessant process of creation, in turn “filling” the dreamtime
with new historical processes. This all-encompassing spiritual
fabric explains how everything came into being and explains
the unwritten laws by which the Aborigines live. According
to their belief, the events of the dreamtime manifest themselves
in landmarks such as rocks, springs and other natural phenomena.
```

I don't mean research for a project or something else specifically scheduled. When I just read articles on the web, I don't care about the publication dates. Whether the article is from 1997 or from the current year 2019. The information is not bound to real time.

The window design problem

If we normally surf the net we can do this with different hardware. Many were on the road in the 90' with a tube monitor and that hasn't changed for a long time. From the year 2000, laptops with a landscape format (the distance of height is shorter than that of length) were currently used by users inside. Nowadays you can use almost everything. Desktops, laptops, e-readers, tablets and smartphones. I think it's a rumor that someone looks at websites on a smartwatch. In any case, this is a great challenge for web developers and designers. The best way is to make the website readable in all formats, because you want to acquire as many customers as possible.

Now I have examined how my own page is displayed in the browser window (in a large and a small window), as well as in the RSS feed. In the picture with the two windows I should actually show much more in the browser, because I have quite a lot of white space that is not used. Even if I have my website displayed in a large window, a lot of space is wasted. Now I have deliberately chosen the format in the last example, because I have no interest in displaying sentences that are much too long. I use a width that shows about 15 words side by side. I can grasp the paragraphs quickly while reading and do not jump by a coincidence into a wrong line. Most of the websites I tested (without CSS formatting) show up to 35 words.

If we pay attention to this small detail, wouldn't it be better if the browser windows were displayed in a portrait format? Apparently we are used to that from books and magazines and it would be an advantage if we took over this artifact, wouldn't it? Because even a magazine like the Wired attaches more importance to a portrait format and does without a filling design. The format is automatically adapted to the window and simply centered if the width is exceeded. At first glance, this is a good solution, because the basic problem remains. Should we not strive for a uniform window format?

The problem with our current monitors is that the format is not designed for the web. First of all it is built like a desk, then like a cinema screen or a work surface where you can place your tools to the right and left. Computer games can also be played better with a landscape format, because the players have become accustomed to the format. Especially in third person games where you have to capture a large part of the game action in order to be able to process information quickly as a

human being, this is a huge advantage. The web has been overlooked because nothing has been optimized for reading. Like here, I now exclude browser games and video portals. It's about the reading.

I often read the pdf format in the browser, as scientific publications or pdf books (Silvanus P. Thompson, F.R.S., Calculus Made Easy, New York, The Macmillan Company, "6-14-43") are usually published in this format. When I have the whole height of the single book page displayed, the font is very small and you can see again the wasted space. If I have the page displayed in full width, the font is unnaturally large and I can't see the whole content at a glance. That's what I do subconsciously. I look at the complete page for a few seconds and capture the large image, then I start to extract the details from the text that I need for my work. Now one could argue that the tablet is a solution to this problem. Unfortunately it isn't, because the tablet was invented for an entertainment web, not for a content web. When I read pdf, I also have to take notes, do quick research and use another input device. Usually this is a keyboard and mouse. I only read novels and other light fiction as paperback.

I tried several times to use a digital keyboard on a tablet productively, but unfortunately I did not succeed. You can chat a little, write small sections of text or even write a longer mail but more is not possible. Developing a Python script on a tablet sensibly and in a reasonable amount of time seems to me to be unproductive. So there should be a way to switch between portrait and landscape format. This concerns the hardware in this case and I will need to gather more information before we get to this section.

Why we have to re:design RSS

In the last weeks there have been some movements about RSS. Mozilla Firefox has decided not to support this file format in future versions of its browser. At first glance, this is a natural development of software that simply takes modules that are not used by users inside out of the program and stops development. As a computer scientist I can understand and support Mozilla's decision. RSS still doesn't seem to be used properly even in large magazines and there seem to be quite a few problems to deal with. Even though I have already mentioned this in similar thoughts in some of the paragraphs above, I would like to reiterate some of the points here.

The design in the browser sucks.

Of course, most people use RSS in an RSS reader software, such as Feedly or other feed aggregators. Software developers (mostly) make an effort when publishing a product. This ensures that readers that either have a design adapted to the target group (feedly) or are logically structured (tiny tiny RSS runs in the browser) survive best. If you still come across an interesting blog or news page while surfing the net and click on the feed link you will always be rewarded with an ugly design. I will not go into this again, as I have already described it in detail. When we look at the landing page of a typical Mastodon user, we see quite a few similarities to a rss.xml file. We have an image of an avatar (you can also add it to RSS) and some additional information. Then come the individual entries (which are limited to 500 characters) for Mastodon. All in all, a Mastodon site is RSS only in a better design.

The RSS syntax is based on XML

The syntax of RSS is based on the XML and is therefore disqualified in all aspects of user-friendliness. In recent years, Markdown has introduced similar languages into web development because in most cases they can simply be read better by people. There used to be the crazy idea of making websites legible for machines instead of people. Really crazy. Personally, I consider the use of angle brackets to be unnecessary, because they prevent the required readability. Below we see an example.

Of course there were also ideas (Brent Simmons & Manton Reece) how to translate the syntax into a readable format. Their idea is based on the JSON file format and called the project JSON Feed Version 1. And even if this idea is already a lot better than its predecessor, there are still some difficulties. JSON is something used by programmers. The mass of normal untrained users inside cannot use this, because they want to create a feed and publish information as fast as possible. Open and close open and curved brackets. Why? Can't you just write that? These are questions the typical user asks himself. So you have to find a format that is not only complex in its application possibilities, but can be learned easily and quickly.

```
{
   "version": "https://jsonfeed.org/version/1",
   "title": "My Example Feed",
   "home_page_url": "https://example.org/",
   "feed_url": "https://example.org/feed.json",
   "items": [
   {
         "id": "2",
         "content_text": "This is a second item.",
         "url": "https://example.org/second-item"
   },
   {
         "id": "1",
         "content_html": "Hello, world!",
```

```
            "url": "https://example.org/initial-post"
        }
    ]
}
```

So we not only need to find a syntax that is used by many people and also progresses in development, but is also easy for people to understand. For this we have to consider the basic needs and leave our echo chamber of the developer, because it must not be a project that does not work in reality and when we look at RSS, this seems to be the case in many points. But it already has many advantages. It is easy to implement into existing software, it is open and information can spread incredibly quickly. These are the hooks that we have to re-lubricate and repair in some places.

Digital evolution

There are many ways to destroy your online business. Some rely on poor customer service, or inadmissible services or simply lousy products. Unfortunately, as we will soon see, there are other ways to take your business to the wall. For example, if you can't access the website. But the reason doesn't matter, mostly it's just trivialities. So in the 90' Flash applications were quite popular. Unfortunately, the company Adobe is not known to close security holes and so the Internet has quietly and secretly agreed to simply exchange everything with HTML5. There was also Java Web Start once, but nobody really talks about it anymore who is seriously still on clear understanding.

Back then. In the time of the information web (see above) it was still possible simply to load an important text file on a server to which then everyone with different software could access. As an open and very flexible file format, this is the beginning of digital evolution. It all started with it and even if you wish it otherwise, this format is still popular and up-to-date on the net. You can open it with almost any program without having to install anything first. But since this was not enough for the entertainment web, more and more new formats were added. HTML is a very important development, because without it one could not imagine the web. To make everything look better, CSS was added at some point, Javascript was developed for more functions and interaction with the user. More and more. Many websites you visit nowadays have become heavy, sluggish and useless. In the digital evolution, what is fastest and most convenient for the user is gaining acceptance. Advertising. Makes the site slow and that is why the supplier market of adblockers like Ublock origin could grow so strongly. Tracking Scripts, Cockies and additional bother to your customer will let him switch to another provider.

The competition is big in the digital evolution and the weak or slow websites are eventually overtaken by the small projects. Customers are no longer so patient and every error is tagged with a closed tab. I don't care why I can't visit the Los Angeles Times website where I want to find out about current topics for a future vacation. As a user, I don't even know what the GDPR is. I don't care either. If the site is not accessible, it must be broken and then I simply look for another one. There's probably another news magazine in Los Angeles, or New York or outside Europe. If your page is not accessible, it will not be read. If it is not read, it is unlikely to be linked to other sites. If it is not linked, your project has no relevance for search engines and if people can't find you on search engines, you don't exist. A brand that is not mentioned on the web does not exist. This company was then excluded from the digital evolution and will slowly but surely die out and be overtaken by successors. Your startup can be as fancy as it may be, if you had a Flash site developed for your business in 2018, I would rather not invest any money in you.

Many providers think their users are just stupid and want them to just click on anything they rub under their noses. But you can get a quick picture of the website in the World Wide Web, simply install an adblocker like UBlock Origin or find out by chance what dark patterns are. Since Internet has the incredibly great advantage of researching everything in a few minutes. Of course there is a lot of fake and you shouldn't believe everything, but if I can't use Motherbard's website without clicking on a button, I ask myself as a vigilant internet user: *Why is that?* Of course not everyone does that, but some and these few (like me) write about it and explain it. This enlightenment is linked and with it me and my project are back in-game of digital evolution. I'm available to my readers and word gets around. Be fair and just make a good product and the readers or customers will come back on their own.

Sometimes just bad design is a reason why nobody uses your site. Sure, the most people use Firefox or Google Chrome but there are hundreds of developers and all have their peculiarities. The Alligator browser is based on Safari and Webkit and Python. They have proven to be a good alternative for my development. It is always a stupid idea to follow a current trend (browser), because tastes can change. Web browsers are only one product and are subject to economic demand. If your website appeals to as many people as possible, you can be sure you are on the right side of the Internet. If your designers and programmers rely only on Javascript frameworks like e.g. vue.js, you will inevitably scare away all those who have issued Javascript in their browser.

It is also a bad custom as a provider to impose on its readers. You're supposed to get a coupon here. Please subscribe to a newsletter and much worse. It's a logical thing to do. The more I annoy my reader with mischief, the less he can read or shop. I f I really want to buy a book because I feel like buying a book *absolutely now* I don't want to click popup overlay away, because *I just want to buy a fucking book*. If I can't buy anything because I don't give up patience for clicks, I have a bad

feeling. This bad feeling is linked to your domain name in my brain and guess what happens now. Right. You will be excluded from the digital evolution with your shop. Just like that. I'll go to a another provider. What? At this shop the book is six euro more expensive? It doesn't matter. *I finally bought a fucking goddammit book for it*.

What I want to say is that any hurdle that your reader has to take keeps him from dealing with your offer. A reader doesn't care what Cloudflare is, because you don't see a page. Of course, you cannot customize your service to the GDPR, but then 741,447,158 Europeans will not be able to read your page. Not everyone knows what an online proxy is or how to install an adblocker. most people just want to find and read information on your site. If you lock the door of your supermarket and wonder why nobody goes shopping in your store, you should perhaps do a retraining. You will find it very difficult to earn money with this kind of business.

The Social Media Desaster

There are many reasons not to register with a commercial social network. We don't need to talk about that. Unfortunately, there are similar issues with free networks like Mastodon. If you register with an instance, admins can kick you again. Whether you like it or not, you are also exposed to the will of other people with this solution. It doesn't even have to have a good reason. In the last five years, the net (and our society) has evolved into a place of redemption that is hard to bear even on very good ideas like Mastodon. I was a short time on an instance on the way and let it be again after a few weeks. It's just hate shit, too, only in a different color and with glitter on it. Of course you can also set up your own instance and is then your own admin, you can also study computer science in the evening. It comes close from the requirement already. Besides one must look now how this develops with the upload filters in Europe and how far the private Admins concerns. The uncertainty remains.

Nevertheless I like the Twitter/Mastodon design. You have a header picture, an avatar and some information about the user to get a first impression quickly. I like that. Below that the entries are displayed in chronological order. I called the old Mastodon page from me and rebuilt it very roughly with the debug inspector as a mockup. This is just an example and should not be a stolen design (I am a programmer, not a designer). For example, I would have divided my feed into text, podcast and media. By media I mean pictures, screenshots and memes. Stuff I post without text. I noticed then on Twitter that I'd often glicked through the pictures just to get sprinkled with $hitpost pictures. Media can also be videos or Bandcamp recommendations. Podcasts I took now only to represent a regular appearing format (e.g. once a month). Every person has his own folder, through Informatik Voodo you can later download individual posts and repost them in his feed. If a feed is closed, at least the interesting posts have spread further. I already mentioned this topic in the chapter *digital evolution*. But this has to be so simple technically that Humans can understand it as well. This could defuse the entire explosive effect that has accumulated in the social media. That's how I imagine it in theory, but it also works in practice ¯\(□)/¯. I also want to bring the web to the desktop, somehow.

Why I will never read your Medium Blog Post

Medium blog posts appear again and again in the feed of news.ycombinator and every time I ignore them. Why? First of all the design of Medium is incredibly exhausting, because they have not only inserted a static header but also a footer. So a part of the content has already been cut away in advance. One is constantly harassed with hints that the readers* inside should move to login to Medium. But I don't want to have to log in just to read a single article. I only comment in very rare cases anyway. I can understand Dave Winer (who is just as stupid and posts his displeasure in another datensilo like twitter) that Tim Berners-Lee (the father of www) could have at least taken the trouble to set up a small one pager. I also think the same about other bloggers. There are enough static frameworks that are really easy to install and if you can't do that you can still use Pastebin. Good content doesn't need design. Medium is a datensilo and should the provider drift into insolvency your articles will be dragged along. To sum it up, I don't care 100% whether you are voted number one on news.ycombinator, because I'd rather cut out my eyes than read a medium blog post and I know what I'm talking about, because I blogged there myself once for a test and was more than dissatisfied after a few weeks.

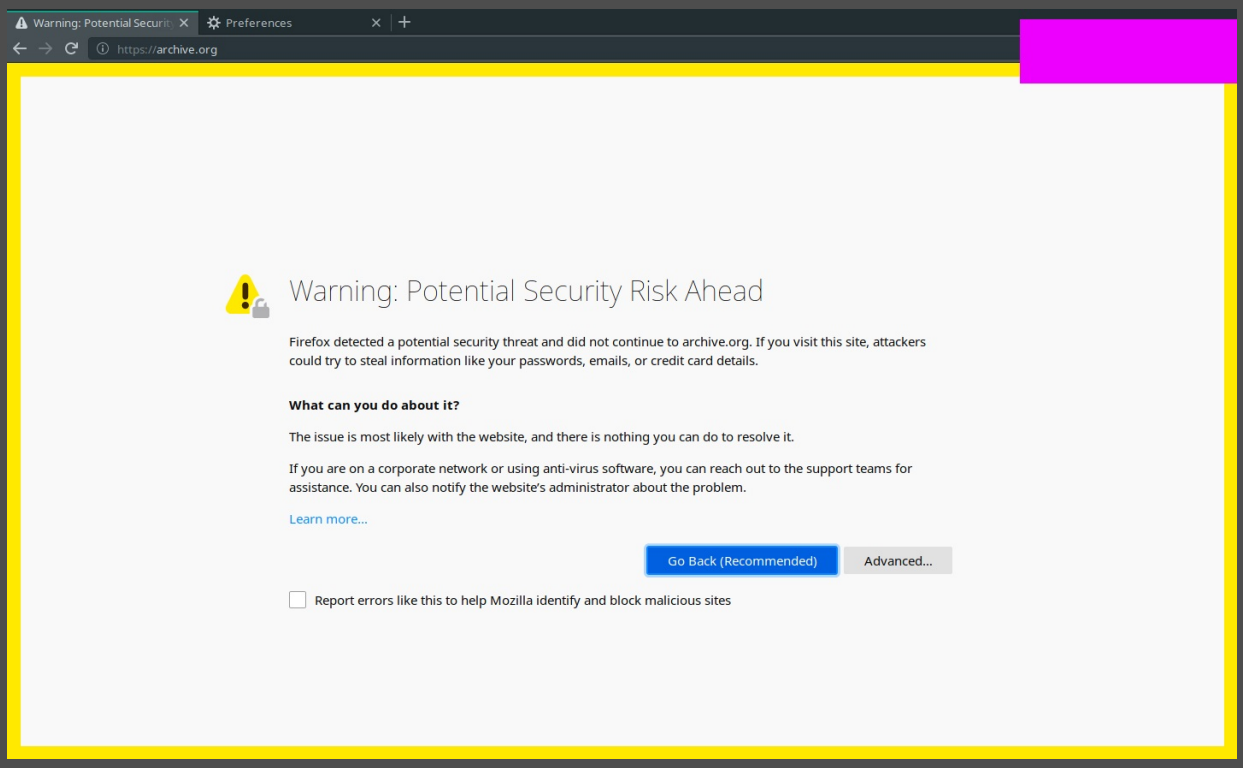
The new Copyright Web

As Julia Reda has already said, yesterday was a black day for the Internet. In the European Parliament, two laws were waved through that will fundamentally change the web in Europe. The upload filters will oblige all services, startups, forums, etc. to check a filter for possible copyright infringements before uploading. There are so many negative points that I refer to the pdf. The big german publishers and Services have infiltrated the whole European politics with more than unclean methods and forced their plan on them. First and foremost as the leader Axel Voss, with his CDU/CSU and the EPP, has already been held responsible for the GDPR. Technically this delusion of publishing houses, German politics and lobbyists is not possible. An upload filter cannot distinguish between meme, satire and terror propaganda. Even if politicians like to talk themselves into being informed about technical developments, I am better informed as a computer scientist. I am working intensively on this topic. There are many people trying to save the web, but sometimes I wonder if it's better to build something new. Something only technicians and young people understand and live for a free internet. Something you can use together. Without

censorship, upload filters, rules that nobody follows anyway. I think that the community within the internet itself knows best which rules are needed and which are not.

This development is also the reason why I no longer link to German newspapers, articles or websites. I simply lock this material out and leave the rest to digital evolution. This is also the reason why I save my website at Gitlab abroad and no longer within the European Union on a German server. This is the reason why I only publish my articles in English and not in my native language German. The European politicians are destroying the web/internet so that publishers who have shot themselves in the knees with the ancillary copyright for press publishers can survive halfway. Print is dead and the artificial intelligence fake news algorithm kills the journalist media star. I deliberately use UBlock Origin and my own filter. Of course the web would be much nicer, if I could do without it, but so far it looks like a race and there will only be losers.

Browser as Gatekeeper



When Tim Berners Lee developed the first web browser in 1990 and laid the foundation stone for today's web, he certainly could not have imagined the rapid development that this technology would go through. At that time it was only about displaying websites and `html` files over longer distances. A scientist in Switzerland should be able to view the research results of a colleague in the Netherlands. That was already the basic idea of the Internet, but it should all look a little better. Unfortunately Google and the other browser manufacturers have developed themselves as gatekeepers for whom it is only important to expand their own position on the software market. Hundreds of lobbyists in consortia, research groups, etc. get inspired to promote their own developments. Unfortunately then open developments like RSS break away, because these are simply no longer supported by the big browser manufacturers. I use a window adapted to me into the digital environment and users should always keep this in mind. Of course you can always trick the settings or just develop your own browser. Already now we see a partially censored can network and that will not change in the next years, when some developers get together and finally develop a really free and for users easy to configure web browser.